# Assignment 3.2

**Name: Shikha Singh**

**Roll no: 25**

**Note**: Use try with resources to handle exceptions and close the resources in all questions except Q2.

Q1. WAP to demonstrate the constructors and methods of File class.

```
import java.io.File;

import java.io.IOException;



public class FileExample {

    public static void main(String[] args) {

        // 1. Creating File object using absolute path

        File file1 = new File("C:/example/testfile.txt");



        // 2. Creating File object using relative path

        File file2 = new File("testfile.txt");
```

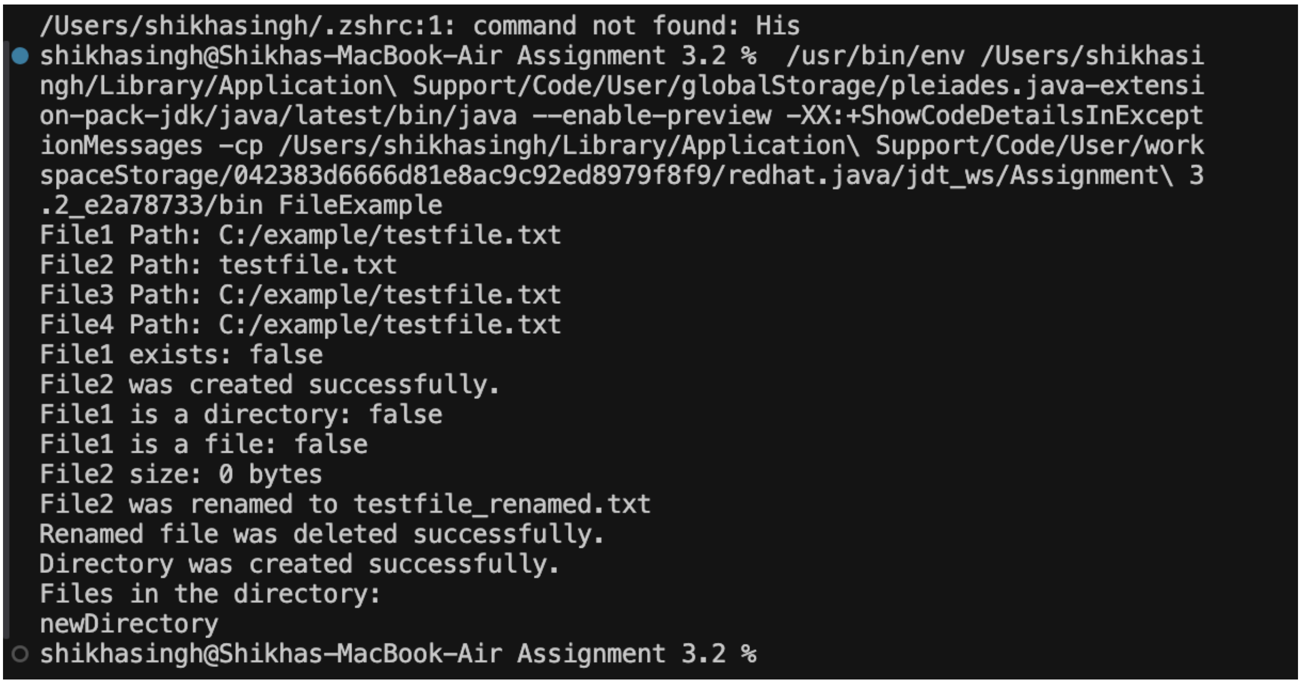
```
// 3. Creating File object using directory and file name
File file3 = new File("C:/example", "testfile.txt");

// 4. Creating File object using parent File object and file name
File parentDir = new File("C:/example");
File file4 = new File(parentDir, "testfile.txt");

// Display file paths
System.out.println("File1 Path: " + file1.getPath());
System.out.println("File2 Path: " + file2.getPath());
System.out.println("File3 Path: " + file3.getPath());
System.out.println("File4 Path: " + file4.getPath());
```

```
}

// Checking if file exists

System.out.println("File1 exists: " + file1.exists());



// Creating a new file

try {

    if (file2.createNewFile()) {

        System.out.println("File2 was created successfully.");

    } else {

        System.out.println("File2 already exists.");

    }

} catch (IOException e) {

    System.out.println("An error occurred while creating File2.");

    e.printStackTrace();

}
```

```
// Checking if it is a directory

System.out.println("File1 is a directory: " + file1.isDirectory());



// Checking if it is a file

System.out.println("File1 is a file: " + file1.isFile());



// Getting file size

System.out.println("File2 size: " + file2.length() + " bytes");



// Renaming the file

File renamedFile = new File("testfile_renamed.txt");

if (file2.renameTo(renamedFile)) {

    System.out.println("File2 was renamed to testfile_renamed.txt");
```

```
} else {
    System.out.println("Renaming failed.");
}

// Deleting the file
if (renamedFile.delete()) {
    System.out.println("Renamed file was deleted successfully.");
} else {
    System.out.println("Failed to delete the renamed file.");
}

// Creating a directory
File directory = new File("C:/example/newDirectory");
if (directory.mkdirs()) {
    System.out.println("Directory was created successfully.");
} else {
    System.out.println("Directory already exists or creation failed.");
}


// Listing files in a directory
File dir = new File("C:/example");
if (dir.isDirectory()) {
    String[] files = dir.list();
    if (files != null) {
        System.out.println("Files in the directory:");
        for (String fileName : files) {
            System.out.println(fileName);
        }
```

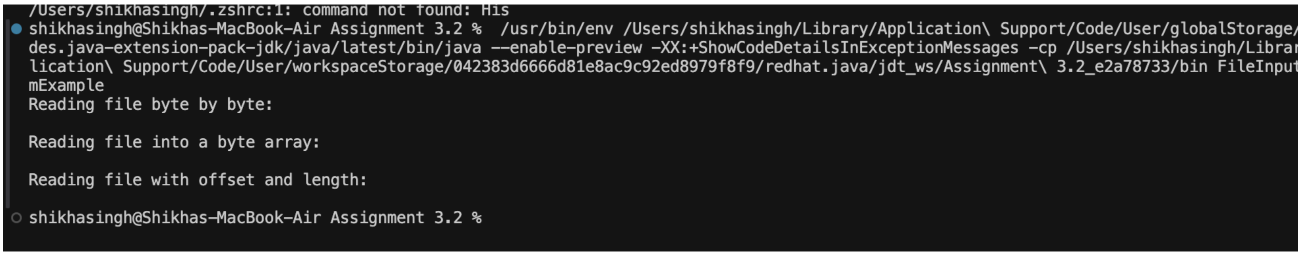
```
            }

        }

    }

}
```



```
/Users/shikhasingh/.zshrc:1: command not found: His
shikhasingh@Shikhas-MacBook-Air Assignment 3.2 %  /usr/bin/env /Users/shikhasi
ngh/Library/Application\ Support/Code/User/globalStorage/pleiades.java-extensi
on-pack-jdk/java/latest/bin/java --enable-preview -XX:+ShowCodeDetailsInExcept
ionMessages -cp /Users/shikhasingh/Library/Application\ Support/Code/User/work
spaceStorage/042383d6666d81e8ac9c92ed8979f8f9/redhat.java/jdt_ws/Assignment\ 3
.2_e2a78733/bin FileExample
File1 Path: C:/example/testfile.txt
File2 Path: testfile.txt
File3 Path: C:/example/testfile.txt
File4 Path: C:/example/testfile.txt
File1 exists: false
File2 was created successfully.
File1 is a directory: false
File1 is a file: false
File2 size: 0 bytes
File2 was renamed to testfile_renamed.txt
Renamed file was deleted successfully.
Directory was created successfully.
Files in the directory:
newDirectory
shikhasingh@Shikhas-MacBook-Air Assignment 3.2 %
```

Q2. WAP to demonstrate how to read a file using FileInputStream using different read methods (read(), read(byte b[], read(byte b[], int off, int len). Use try, catch finally to handle exceptions and close the resources.

```
import java.io.FileInputStream;
import java.io.IOException;
public class FileInputStreamExample {
    public static void main(String[] args) {
        String filePath = "/Users/shikhasingh/Desktop/Assignment 3.2/example.txt";
        FileInputStream fileInputStream = null;
        try {

            fileInputStream = new FileInputStream(filePath);

            System.out.println("Reading file byte by byte:");
            int byteData;
            while ((byteData = fileInputStream.read()) != -1) {
                System.out.print((char) byteData);
            }
            System.out.println();

            fileInputStream.close();
            fileInputStream = new FileInputStream(filePath);

            System.out.println("Reading file into a byte array:");
            byte[] byteArray = new byte[1024];
            int bytesRead = fileInputStream.read(byteArray);
            if (bytesRead != -1) {
                System.out.write(byteArray, 0, bytesRead);
            }
            System.out.println();

            fileInputStream.close();
            fileInputStream = new FileInputStream(filePath);

            System.out.println("Reading file with offset and length:");
            byteArray = new byte[1024];
            int offset = 10;
            int length = 50;
            bytesRead = fileInputStream.read(byteArray, offset, length);
            if (bytesRead != -1) {
                System.out.write(byteArray, offset, bytesRead);
            }
            System.out.println();
        } catch (IOException e) {
            System.out.println("An error occurred while reading the file.");
            e.printStackTrace();
        } finally {

            if (fileInputStream != null) {
                try {
                    fileInputStream.close();
                } catch (IOException e) {
                    System.out.println("An error occurred while closing the fileInputStream.");
                    e.printStackTrace();
                }
            }
        }
    }
}
```

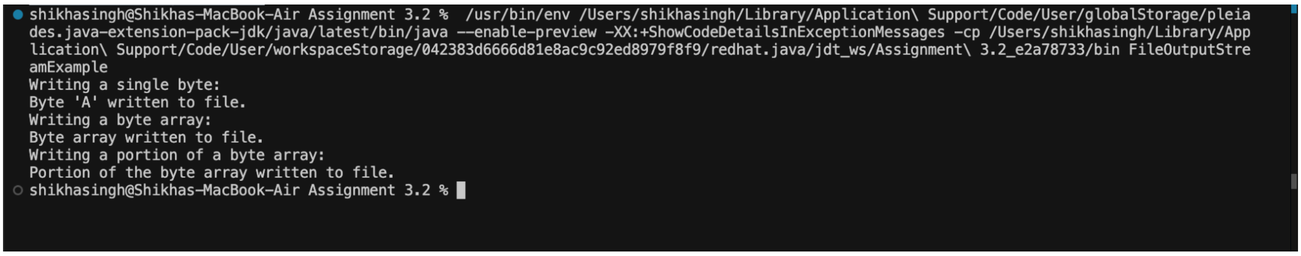


```
/Users/shikhasingh/.zshrc:1: command not found: His
shikhasingh@Shikhas-MacBook-Air Assignment 3.2 %  /usr/bin/env /Users/shikhasingh/Library/Application\ Support/Code/User/globalStorage/
des.java-extension-pack-jdk/java/latest/bin/java --enable-preview -XX:+ShowCodeDetailsInExceptionMessages -cp /Users/shikhasingh/Librar
lication\ Support/Code/User/workspaceStorage/042383d6666d81e8ac9c92ed8979f8f9/redhat.java/jdt_ws/Assignment\ 3.2_e2a78733/bin FileInput
mExample
Reading file byte by byte:

Reading file into a byte array:

Reading file with offset and length:

shikhasingh@Shikhas-MacBook-Air Assignment 3.2 %
```

## Q3. WAP to demonstrate how to write a file using FileOutputStream using different write methods.

```
import java.io.FileOutputStream;
import java.io.IOException;
public class FileOutputStreamExample {
   public static void main(String[] args) {

      String filePath = "/Users/shikhasingh/Desktop/Assignment 3.2/output.txt";
      FileOutputStream fileOutputStream = null;
      try {

         fileOutputStream = new FileOutputStream(filePath);

         System.out.println("Writing a single byte:");
         fileOutputStream.write(65);
         System.out.println("Byte 'A' written to file.");

         System.out.println("Writing a byte array:");
         byte[] byteArray = "Hello, FileOutputStream!".getBytes();
         fileOutputStream.write(byteArray);
         System.out.println("Byte array written to file.");

         System.out.println("Writing a portion of a byte array:");
         byteArray = "This is a portion of the array.".getBytes();
         int offset = 5;
         int length = 10;
         fileOutputStream.write(byteArray, offset, length);
         System.out.println("Portion of the byte array written to file.");
      } catch (IOException e) {
         System.out.println("An error occurred while writing to the file.");
         e.printStackTrace();
      } finally {

         if (fileOutputStream != null) {
            try {
               fileOutputStream.close();
            } catch (IOException e) {
               System.out.println("An error occurred while closing the fileOutputStream.");
               e.printStackTrace();
            }
         }
```

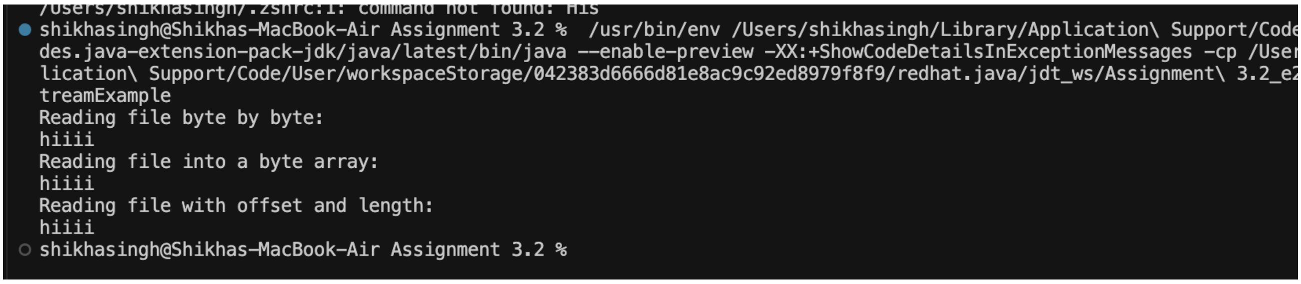
```
        }
    }
}
```



```
● shikhasingh@Shikhas-MacBook-Air Assignment 3.2 %  /usr/bin/env /Users/shikhasingh/Library/Application\ Support/Code/User/globalStorage/pleia
  des.java-extension-pack-jdk/java/latest/bin/java --enable-preview -XX:+ShowCodeDetailsInExceptionMessages -cp /Users/shikhasingh/Library/App
  lication\ Support/Code/User/workspaceStorage/042383d6666d81e8ac9c92ed8979f8f9/redhat.java/jdt_ws/Assignment\ 3.2_e2a78733/bin FileOutputStre
  amExample
  Writing a single byte:
  Byte 'A' written to file.
  Writing a byte array:
  Byte array written to file.
  Writing a portion of a byte array:
  Portion of the byte array written to file.
○ shikhasingh@Shikhas-MacBook-Air Assignment 3.2 % 
```

## Q4. WAP to demonstrate how to read a file using BufferedInputStream.

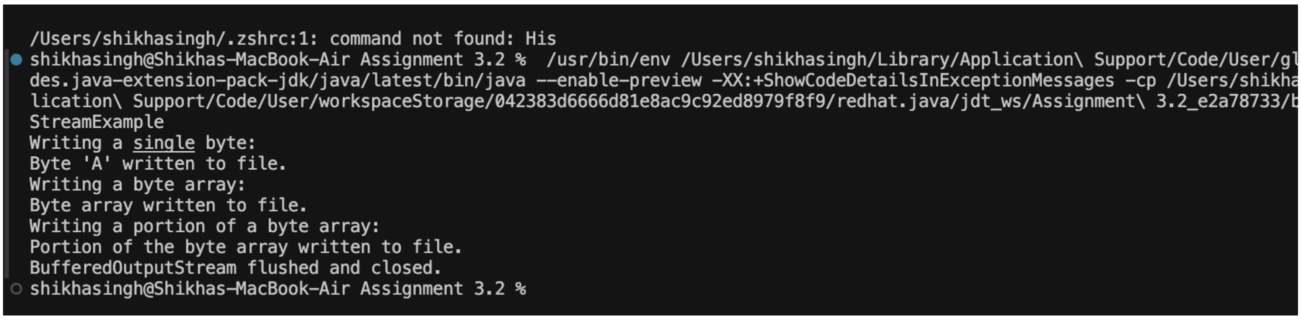
```
import java.io.BufferedInputStream;
import java.io.FileInputStream;
import java.io.IOException;
public class BufferedInputStreamExample {
    public static void main(String[] args) {
        // Specify the path to the file you want to read from
        String filePath = "/Users/shikhasingh/Desktop/Assignment 3.2/input.txt";
        BufferedInputStream bufferedInputStream = null;
        try {
            // Create a FileInputStream object
            FileInputStream fileInputStream = new FileInputStream(filePath);
            // Wrap FileInputStream with BufferedInputStream
            bufferedInputStream = new BufferedInputStream(fileInputStream);
            // 1. Read file byte by byte using read() method
            System.out.println("Reading file byte by byte:");
            int byteData;
            while ((byteData = bufferedInputStream.read()) != -1) {
                System.out.print((char) byteData);
            }
            System.out.println();
            // Reset BufferedInputStream to the beginning of the file
            bufferedInputStream.close();
            fileInputStream = new FileInputStream(filePath);
            bufferedInputStream = new BufferedInputStream(fileInputStream);
            // 2. Read file into a byte array using read(byte b[]) method
            System.out.println("Reading file into a byte array:");
            byte[] byteArray = new byte[1024]; // Buffer size of 1024 bytes
            int bytesRead = bufferedInputStream.read(byteArray);
            if (bytesRead != -1) {
                System.out.write(byteArray, 0, bytesRead); // Writing to standard output
            }
            System.out.println();
            // Reset BufferedInputStream to the beginning of the file
            bufferedInputStream.close();
            fileInputStream = new FileInputStream(filePath);
            bufferedInputStream = new BufferedInputStream(fileInputStream);
            // 3. Read file into a byte array with offset and length using read(byte b[], int off, int len) method
            System.out.println("Reading file with offset and length:");
            byteArray = new byte[1024]; // Buffer size of 1024 bytes
            int offset = 10; // Starting from the 10th byte
            int length = 50; // Reading 50 bytes
            bytesRead = bufferedInputStream.read(byteArray, offset, length);
            if (bytesRead != -1) {
                System.out.write(byteArray, offset, bytesRead); // Writing to standard output
            }
            System.out.println();
        } catch (IOException e) {
            System.out.println("An error occurred while reading the file.");
            e.printStackTrace();
        } finally {
            // Ensure that the bufferedInputStream is closed
            if (bufferedInputStream != null) {
                try {
                    bufferedInputStream.close();
                } catch (IOException e) {
                    System.out.println("An error occurred while closing the bufferedInputStream.");
                    e.printStackTrace();
                }
            }
        }
    }
}
```



```
/Users/shikhasingh/.zshrc:1: command not found: His
● shikhasingh@Shikhas-MacBook-Air Assignment 3.2 %  /usr/bin/env /Users/shikhasingh/Library/Application\ Support/Code
  des.java-extension-pack-jdk/java/latest/bin/java --enable-preview -XX:+ShowCodeDetailsInExceptionMessages -cp /User
  lication\ Support/Code/User/workspaceStorage/042383d6666d81e8ac9c92ed8979f8f9/redhat.java/jdt_ws/Assignment\ 3.2_e2
  treamExample
  Reading file byte by byte:
  hiiii
  Reading file into a byte array:
  hiiii
  Reading file with offset and length:
  hiiii
○ shikhasingh@Shikhas-MacBook-Air Assignment 3.2 %
```

## Q5. WAP to demonstrate how to write a file using BufferedOutputStream.

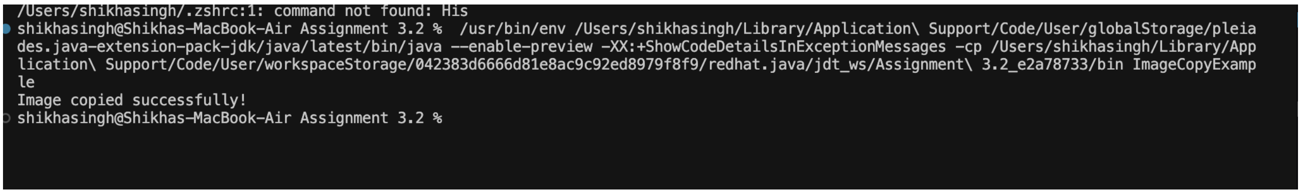
```
import java.io.BufferedOutputStream;
import java.io.FileOutputStream;
import java.io.IOException;
public class BufferedOutputStreamExample {
    public static void main(String[] args) {

        String filePath = "/Users/shikhasingh/Desktop/Assignment 3.2/output.txt";
        BufferedOutputStream bufferedOutputStream = null;
        try {

            FileOutputStream fileOutputStream = new FileOutputStream(filePath);

            bufferedOutputStream = new BufferedOutputStream(fileOutputStream);

            System.out.println("Writing a single byte:");
            bufferedOutputStream.write(65);
            System.out.println("Byte 'A' written to file.");

            System.out.println("Writing a byte array:");
            byte[] byteArray = "Hello, BufferedOutputStream!".getBytes();
            bufferedOutputStream.write(byteArray);
            System.out.println("Byte array written to file.");

            System.out.println("Writing a portion of a byte array:");
            byteArray = "This is a portion of the array.".getBytes();
            int offset = 5;
            int length = 10;
            bufferedOutputStream.write(byteArray, offset, length);
            System.out.println("Portion of the byte array written to file.");

            bufferedOutputStream.flush();
            System.out.println("BufferedOutputStream flushed and closed.");
        } catch (IOException e) {
            System.out.println("An error occurred while writing to the file.");
            e.printStackTrace();
        } finally {

            if (bufferedOutputStream != null) {
                try {
                    bufferedOutputStream.close();
                } catch (IOException e) {
                    System.out.println("An error occurred while closing the bufferedOutputStream.");
                    e.printStackTrace();
                }
            }
        }
    }
}
```



```
/Users/shikhasingh/.zshrc:1: command not found: His
shikhasingh@Shikhas-MacBook-Air Assignment 3.2 %  /usr/bin/env /Users/shikhasingh/Library/Application\ Support/Code/User/gl
des.java-extension-pack-jdk/java/latest/bin/java --enable-preview -XX:+ShowCodeDetailsInExceptionMessages -cp /Users/shikha
lication\ Support/Code/User/workspaceStorage/042383d6666d81e8ac9c92ed8979f8f9/redhat.java/jdt_ws/Assignment\ 3.2_e2a78733/b
StreamExample
Writing a single byte:
Byte 'A' written to file.
Writing a byte array:
Byte array written to file.
Writing a portion of a byte array:
Portion of the byte array written to file.
BufferedOutputStream flushed and closed.
shikhasingh@Shikhas-MacBook-Air Assignment 3.2 %
```

## Q6. WAP to copy an image file using the concept of InputStream and OutputStream.

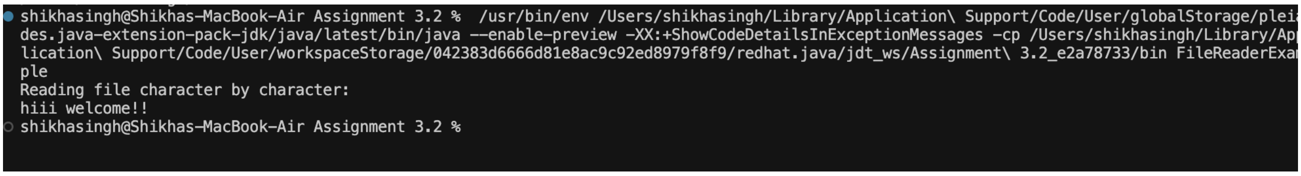
```
import java.io.FileInputStream;
import java.io.FileOutputStream;
import java.io.IOException;
import java.io.InputStream;
import java.io.OutputStream;
public class ImageCopyExample {
    public static void main(String[] args) {

        String sourceFilePath = "/Users/shikhasingh/Desktop/Assignment 3.2/sourceImage.jpg";
        String destinationFilePath = "/Users/shikhasingh/Desktop/Assignment 3.2/destinationImage.jpg";
        InputStream inputStream = null;
        OutputStream outputStream = null;
        try {

            inputStream = new FileInputStream(sourceFilePath);
            outputStream = new FileOutputStream(destinationFilePath);

            byte[] buffer = new byte[1024];
            int bytesRead;

            while ((bytesRead = inputStream.read(buffer)) != -1) {
                outputStream.write(buffer, 0, bytesRead);
            }
            System.out.println("Image copied successfully!");
        } catch (IOException e) {
            System.out.println("An error occurred while copying the image file.");
            e.printStackTrace();
        } finally {

            if (inputStream != null) {
                try {
                    inputStream.close();
                } catch (IOException e) {
                    System.out.println("An error occurred while closing the input stream.");
                    e.printStackTrace();
                }
            }
            if (outputStream != null) {
                try {
                    outputStream.close();
                } catch (IOException e) {
                    System.out.println("An error occurred while closing the output stream.");
                    e.printStackTrace();
                }
            }
        }
    }
}
```



```
/Users/shikhasingh/.zshrc:1: command not found: His
shikhasingh@Shikhas-MacBook-Air Assignment 3.2 %  /usr/bin/env /Users/shikhasingh/Library/Application\ Support/Code/User/globalStorage/pleia
des.java-extension-pack-jdk/java/latest/bin/java --enable-preview -XX:+ShowCodeDetailsInExceptionMessages -cp /Users/shikhasingh/Library/App
lication\ Support/Code/User/workspaceStorage/042383d6666d81e8ac9c92ed8979f8f9/redhat.java/jdt_ws/Assignment\ 3.2_e2a78733/bin ImageCopyExamp
le
Image copied successfully!
shikhasingh@Shikhas-MacBook-Air Assignment 3.2 %
```

## Q7. WAP to demonstrate how to read a text file using FileReader.

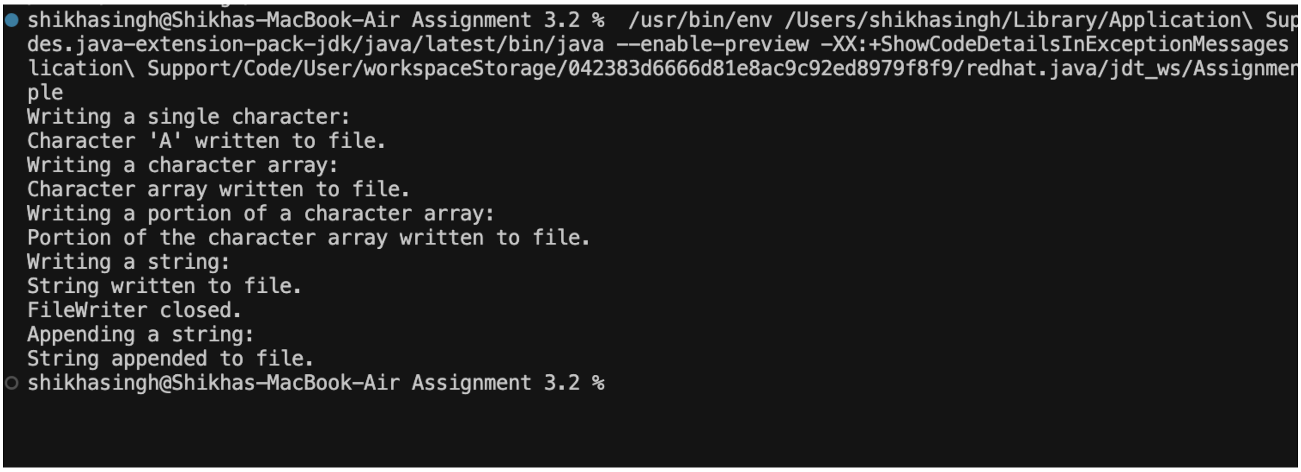
```
import java.io.FileReader;
import java.io.IOException;
public class FileReaderExample {
    public static void main(String[] args) {

        String filePath = "/Users/shikhasingh/Desktop/Assignment 3.2/example.txt";
        FileReader fileReader = null;
        try {
            fileReader = new FileReader(filePath);

            System.out.println("Reading file character by character:");
            int charData;
            while ((charData = fileReader.read()) != -1) {
                System.out.print((char) charData);
            }
            System.out.println();
        } catch (IOException e) {
            System.out.println("An error occurred while reading the file.");
            e.printStackTrace();
        } finally {

            if (fileReader != null) {
                try {
                    fileReader.close();
                } catch (IOException e) {
                    System.out.println("An error occurred while closing the FileReader.");
                    e.printStackTrace();
                }
            }
        }
    }
}
```



```
● shikhasingh@Shikhas-MacBook-Air Assignment 3.2 %  /usr/bin/env /Users/shikhasingh/Library/Application\ Support/Code/User/globalStorage/plei
  des.java-extension-pack-jdk/java/latest/bin/java --enable-preview -XX:+ShowCodeDetailsInExceptionMessages -cp /Users/shikhasingh/Library/Ap
  lication\ Support/Code/User/workspaceStorage/042383d6666d81e8ac9c92ed8979f8f9/redhat.java/jdt_ws/Assignment\ 3.2_e2a78733/bin FileReaderExa
  ple
  Reading file character by character:
  hiii welcome!!
○ shikhasingh@Shikhas-MacBook-Air Assignment 3.2 %
```

## Q8. WAP to demonstrate how to write a text file using FileWriter. Use all the write methods and append methods present in the PPT.

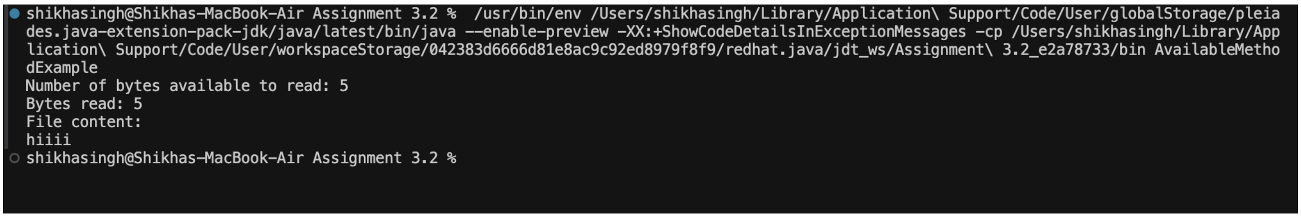
```
import java.io.FileWriter;
import java.io.IOException;
public class FileWriterExample {
    public static void main(String[] args) {

        String filePath = "/Users/shikhasingh/Desktop/Assignment 3.2/output.txt";
        FileWriter fileWriter = null;
        try {

            fileWriter = new FileWriter(filePath);

            System.out.println("Writing a single character:");
            fileWriter.write('A');
            System.out.println("Character 'A' written to file.");

            System.out.println("Writing a character array:");
            char[] charArray = "Hello, FileWriter!".toCharArray();
            fileWriter.write(charArray);
            System.out.println("Character array written to file.");

            System.out.println("Writing a portion of a character array:");
            charArray = "This is a portion of the array.".toCharArray();
            int offset = 5;
            int length = 10;
            fileWriter.write(charArray, offset, length);
            System.out.println("Portion of the character array written to file.");

            System.out.println("Writing a string:");
            fileWriter.write("This is a string written to the file.");
            System.out.println("String written to file.");

            fileWriter.close();
            System.out.println("FileWriter closed.");

            fileWriter = new FileWriter(filePath, true);

            System.out.println("Appending a string:");
            fileWriter.write("\nThis string is appended to the file.");
            System.out.println("String appended to file.");
        } catch (IOException e) {
            System.out.println("An error occurred while writing to the file.");
            e.printStackTrace();
        } finally {

            if (fileWriter != null) {
                try {
                    fileWriter.close();
                } catch (IOException e) {
                    System.out.println("An error occurred while closing the FileWriter.");
                    e.printStackTrace();
                }
            }
        }
    }
}
```



```
● shikhasingh@Shikhas-MacBook-Air Assignment 3.2 %  /usr/bin/env /Users/shikhasingh/Library/Application\ Sup
  des.java-extension-pack-jdk/java/latest/bin/java --enable-preview -XX:+ShowCodeDetailsInExceptionMessages
  lication\ Support/Code/User/workspaceStorage/042383d6666d81e8ac9c92ed8979f8f9/redhat.java/jdt_ws/Assignmen
  ple
  Writing a single character:
  Character 'A' written to file.
  Writing a character array:
  Character array written to file.
  Writing a portion of a character array:
  Portion of the character array written to file.
  Writing a string:
  String written to file.
  FileWriter closed.
  Appending a string:
  String appended to file.
○ shikhasingh@Shikhas-MacBook-Air Assignment 3.2 %
```

## Q9. WAP to demonstrate the use of available() method.

```
import java.io.FileInputStream;
import java.io.IOException;
public class AvailableMethodExample {
    public static void main(String[] args) {

        String filePath = "/Users/shikhasingh/Desktop/Assignment 3.2/input.txt";
        FileInputStream fileInputStream = null;
        try {

            fileInputStream = new FileInputStream(filePath);

            int availableBytes = fileInputStream.available();
            System.out.println("Number of bytes available to read: " + availableBytes);

            byte[] buffer = new byte[availableBytes];
            int bytesRead = fileInputStream.read(buffer);
            System.out.println("Bytes read: " + bytesRead);
            System.out.println("File content:");
            System.out.write(buffer, 0, bytesRead);
            System.out.println();
        } catch (IOException e) {
            System.out.println("An error occurred while reading the file.");
            e.printStackTrace();
        } finally {

            if (fileInputStream != null) {
                try {
                    fileInputStream.close();
                } catch (IOException e) {
                    System.out.println("An error occurred while closing the FileInputStream.");
                    e.printStackTrace();
                }
            }
        }
    }
}
```



```
● shikhasingh@Shikhas-MacBook-Air Assignment 3.2 %  /usr/bin/env /Users/shikhasingh/Library/Application\ Support/Code/User/globalStorage/pleia
  des.java-extension-pack-jdk/java/latest/bin/java --enable-preview -XX:+ShowCodeDetailsInExceptionMessages -cp /Users/shikhasingh/Library/App
  lication\ Support/Code/User/workspaceStorage/042383d6666d81e8ac9c92ed8979f8f9/redhat.java/jdt_ws/Assignment\ 3.2_e2a78733/bin AvailableMetho
  dExample
  Number of bytes available to read: 5
  Bytes read: 5
  File content:
  hiiii
○ shikhasingh@Shikhas-MacBook-Air Assignment 3.2 %
```

## Q10. WAP to demonstrate the use of the following methods:

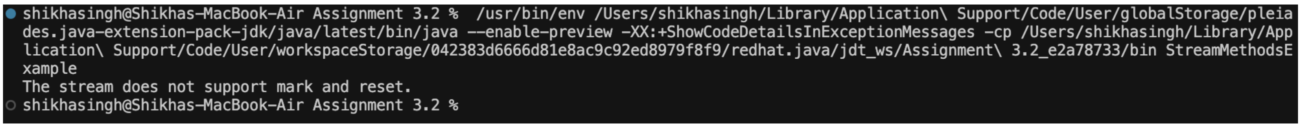
markSupported()

mark()

reset()

skip()

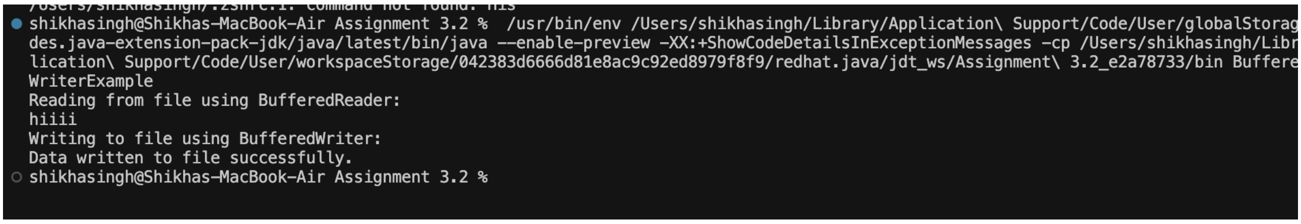
```
import java.io.FileInputStream;
import java.io.IOException;
public class StreamMethodsExample {
    public static void main(String[] args) {

        String filePath = "/Users/shikhasingh/Desktop/Assignment 3.2/input.txt";
        FileInputStream fileInputStream = null;
        try {

            fileInputStream = new FileInputStream(filePath);

            if (fileInputStream.markSupported()) {
                System.out.println("The stream supports mark and reset.");

                byte[] buffer = new byte[10];
                fileInputStream.read(buffer);
                System.out.println("Read bytes: " + new String(buffer));

                fileInputStream.mark(20);

                fileInputStream.read(buffer);
                System.out.println("Read more bytes: " + new String(buffer));

                fileInputStream.skip(5);
                System.out.println("Skipped 5 bytes.");

                fileInputStream.read(buffer);
                System.out.println("Read bytes after skipping: " + new String(buffer));

                fileInputStream.reset();
                System.out.println("Stream reset to the marked position.");

                fileInputStream.read(buffer);
                System.out.println("Read bytes after reset: " + new String(buffer));

            } else {
                System.out.println("The stream does not support mark and reset.");
            }
        } catch (IOException e) {
            System.out.println("An error occurred while processing the file.");
            e.printStackTrace();
        } finally {

            if (fileInputStream != null) {
                try {
                    fileInputStream.close();
                } catch (IOException e) {
                    System.out.println("An error occurred while closing the FileInputStream.");
                    e.printStackTrace();
                }
            }
        }
    }
}
```



```
● shikhasingh@Shikhas-MacBook-Air Assignment 3.2 %  /usr/bin/env /Users/shikhasingh/Library/Application\ Support/Code/User/globalStorage/pleia
  des.java-extension-pack-jdk/java/latest/bin/java --enable-preview -XX:+ShowCodeDetailsInExceptionMessages -cp /Users/shikhasingh/Library/App
  lication\ Support/Code/User/workspaceStorage/042383d6666d81e8ac9c92ed8979f8f9/redhat.java/jdt_ws/Assignment\ 3.2_e2a78733/bin StreamMethodsE
  xample
  The stream does not support mark and reset.
○ shikhasingh@Shikhas-MacBook-Air Assignment 3.2 %
```

## Q11. WAP to demonstrate the use of BufferedReader and BufferedWriter. Use readLine(), new Line(), read(String) methods.

```
import java.io.BufferedReader;
import java.io.BufferedWriter;
import java.io.FileReader;
import java.io.FileWriter;
import java.io.IOException;
public class BufferedReaderWriterExample {
    public static void main(String[] args) {
        String inputFilePath = "/Users/shikhasingh/Desktop/Assignment 3.2/input.txt";
        String outputFilePath = "/Users/shikhasingh/Desktop/Assignment 3.2/output.txt";

        try (BufferedReader bufferedReader = new BufferedReader(new FileReader(inputFilePath))) {
            System.out.println("Reading from file using BufferedReader:");
            String line;
            while ((line = bufferedReader.readLine()) != null) {
                System.out.println(line);
            }
        } catch (IOException e) {
            System.out.println("An error occurred while reading the file.");
            e.printStackTrace();
        }
        try (BufferedWriter bufferedWriter = new BufferedWriter(new FileWriter(outputFilePath))) {
            System.out.println("Writing to file using BufferedWriter:");
            bufferedWriter.write("Hello, BufferedWriter!");
            bufferedWriter.newLine();
            bufferedWriter.write("This is a new line in the file.");
            bufferedWriter.newLine();
            bufferedWriter.write("BufferedWriter makes writing efficient.");
            System.out.println("Data written to file successfully.");
        } catch (IOException e) {
            System.out.println("An error occurred while writing to the file.");
            e.printStackTrace();
        }
    }
}
```



```
/Users/shikhasingh/.zshrc:1: command not found: his
● shikhasingh@Shikhas-MacBook-Air Assignment 3.2 %  /usr/bin/env /Users/shikhasingh/Library/Application\ Support/Code/User/globalStorag
  des.java-extension-pack-jdk/java/latest/bin/java --enable-preview -XX:+ShowCodeDetailsInExceptionMessages -cp /Users/shikhasingh/Libr
  lication\ Support/Code/User/workspaceStorage/042383d6666d81e8ac9c92ed8979f8f9/redhat.java/jdt_ws/Assignment\ 3.2_e2a78733/bin Buffere
  WriterExample
  Reading from file using BufferedReader:
  hiiii
  Writing to file using BufferedWriter:
  Data written to file successfully.
○ shikhasingh@Shikhas-MacBook-Air Assignment 3.2 %
```

## Q12. WAP to demonstrate the serialization and deserialization of objects. Also

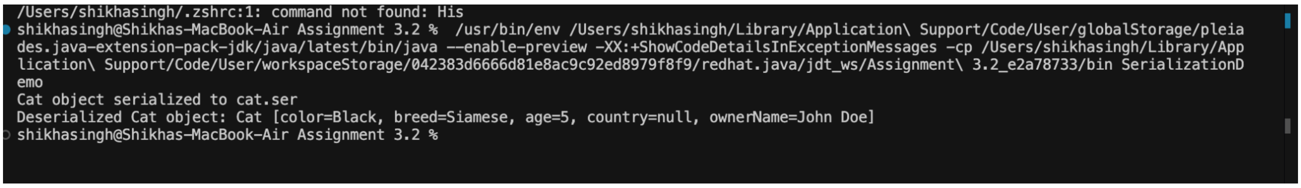
demonstrate transient, static data member for Cat class. Cat class has instance variable: String color, String breed, int age, String country.

## Cat.java

```
import java.io.Serializable;
public class Cat implements Serializable {
   private static final long serialVersionUID = 1L;
   private String color;
   private String breed;
   private int age;
   private transient String country;
   private static String ownerName;

   public Cat(String color, String breed, int age, String country, String ownerName) {
      this.color = color;
      this.breed = breed;
      this.age = age;
      this.country = country;
      Cat.ownerName = ownerName;
   }

   public String getColor() {
      return color;
   }
   public void setColor(String color) {
      this.color = color;
   }
   public String getBreed() {
      return breed;
   }
   public void setBreed(String breed) {
      this.breed = breed;
   }
   public int getAge() {
      return age;
   }
   public void setAge(int age) {
      this.age = age;
   }
   public String getCountry() {
      return country;
   }
   public void setCountry(String country) {
      this.country = country;
   }
   public static String getOwnerName() {
      return ownerName;
   }
   public static void setOwnerName(String ownerName) {
        Cat.ownerName = ownerName;
    }

    public String toString() {
        return "Cat [color=" + color + ", breed=" + breed + ", age=" + age +
               ", country=" + country + ", ownerName=" + ownerName + "]";
    }
}
```

## SerializationDemo.java:

```
import java.io.FileInputStream;
import java.io.FileOutputStream;
import java.io.IOException;
import java.io.ObjectInputStream;
import java.io.ObjectOutputStream;
public class SerializationDemo {
    public static void main(String[] args) {
        String filePath = "cat.ser";

        Cat myCat = new Cat("Black", "Siamese", 5, "USA", "John Doe");

        try (ObjectOutputStream oos = new ObjectOutputStream(new FileOutputStream(filePath))) {
            oos.writeObject(myCat);
            System.out.println("Cat object serialized to " + filePath);
        } catch (IOException e) {
            System.out.println("An error occurred during serialization.");
            e.printStackTrace();
        }

        try (ObjectInputStream ois = new ObjectInputStream(new FileInputStream(filePath))) {
            Cat deserializedCat = (Cat) ois.readObject();
            System.out.println("Deserialized Cat object: " + deserializedCat);
        } catch (IOException | ClassNotFoundException e) {
            System.out.println("An error occurred during deserialization.");
            e.printStackTrace();
        }
    }
}
```



```
/Users/shikhasingh/.zshrc:1: command not found: His
shikhasingh@Shikhas-MacBook-Air Assignment 3.2 %  /usr/bin/env /Users/shikhasingh/Library/Application\ Support/Code/User/globalStorage/pleia
des.java-extension-pack-jdk/java/latest/bin/java --enable-preview -XX:+ShowCodeDetailsInExceptionMessages -cp /Users/shikhasingh/Library/App
lication\ Support/Code/User/workspaceStorage/042383d6666d81e8ac9c92ed8979f8f9/redhat.java/jdt_ws/Assignment\ 3.2_e2a78733/bin SerializationD
emo
Cat object serialized to cat.ser
Deserialized Cat object: Cat [color=Black, breed=Siamese, age=5, country=null, ownerName=John Doe]
shikhasingh@Shikhas-MacBook-Air Assignment 3.2 %
```

## Q13. WAP to demonstrate:

Create an Address class with instance variables of String type:

apptDetails

district

state

country

Create a class Person with instance variables:

int id

String name

Address addr

Create an Employee class and inherit Person class. It’s instance variables:

String EmpId

String Dept

String Designation

You have to serialize and deserialize object of Employee class .

## —Address.java

```
import java.io.Serializable;
public class Address implements Serializable {
   private static final long serialVersionUID = 1L;
   private String apptDetails;
   private String district;
   private String state;
   private String country;

   public Address(String apptDetails, String district, String state, String country) {
      this.apptDetails = apptDetails;
      this.district = district;
      this.state = state;
      this.country = country;
   }

   public String getApptDetails() {
      return apptDetails;
   }
   public void setApptDetails(String apptDetails) {
      this.apptDetails = apptDetails;
   }
   public String getDistrict() {
      return district;
   }
   public void setDistrict(String district) {
      this.district = district;
   }
   public String getState() {
      return state;
```

```
    }
    public void setState(String state) {
        this.state = state;
    }
    public String getCountry() {
        return country;
    }
    public void setCountry(String country) {
        this.country = country;
    }
    @Override
    public String toString() {
        return "Address [apptDetails=" + apptDetails + ", district=" + district + ", state=" + state + ", country=" + country +
"]";
    }
}
```

## —Person.java

```
import java.io.Serializable;
public class Person implements Serializable {
    private static final long serialVersionUID = 1L;
    private int id;
    private String name;
    private Address addr;

    public Person(int id, String name, Address addr) {
        this.id = id;
        this.name = name;
        this.addr = addr;
    }

    public int getId() {
        return id;
    }
    public void setId(int id) {
        this.id = id;
    }
    public String getName() {
        return name;
    }
    public void setName(String name) {
        this.name = name;
    }
    public Address getAddr() {
        return addr;
    }
    public void setAddr(Address addr) {
        this.addr = addr;
    }
    @Override
    public String toString() {
        return "Person [id=" + id + ", name=" + name + ", addr=" + addr + "]";
```

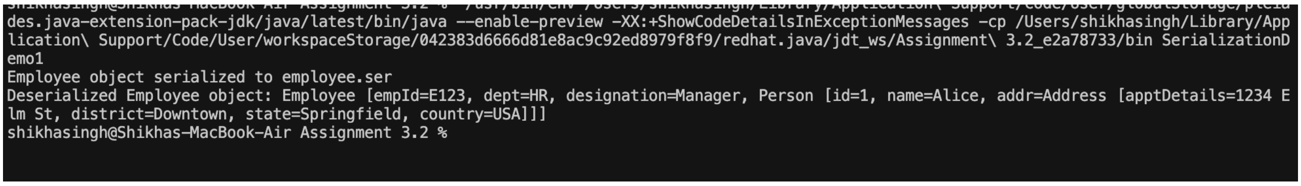
```
  }
}
```

## —Employee.java

```
import java.io.Serializable;
public class Employee extends Person implements Serializable {
  private static final long serialVersionUID = 1L;
  private String empId;
  private String dept;
  private String designation;

  public Employee(int id, String name, Address addr, String empId, String dept, String designation) {
    super(id, name, addr); // Call superclass constructor
    this.empId = empId;
    this.dept = dept;
    this.designation = designation;
  }
  // Getters and Setters
  public String getEmpId() {
    return empId;
  }
  public void setEmpId(String empId) {
    this.empId = empId;
  }
  public String getDept() {
    return dept;
  }
  public void setDept(String dept) {
    this.dept = dept;
  }
  public String getDesignation() {
    return designation;
  }
  public void setDesignation(String designation) {
    this.designation = designation;
  }
  @Override
  public String toString() {
    return "Employee [empId=" + empId + ", dept=" + dept + ", designation=" + designation + ", " + super.toString() +
"]";
  }
}
```

## —SerializationDemo.java

```
import java.io.FileOutputStream;
import java.io.FileInputStream;
import java.io.IOException;
import java.io.ObjectOutputStream;
import java.io.ObjectInputStream;
public class SerializationDemo {
  public static void main(String[] args) {
        String filePath = "employee.ser";

        Address address = new Address("1234 Elm St", "Downtown", "Springfield", "USA");

        Employee employee = new Employee(1, "Alice", address, "E123", "HR", "Manager");

        try (ObjectOutputStream oos = new ObjectOutputStream(new FileOutputStream(filePath))) {
            oos.writeObject(employee);
            System.out.println("Employee object serialized to " + filePath);
        } catch (IOException e) {
            System.out.println("An error occurred during serialization.");
            e.printStackTrace();
        }

        try (ObjectInputStream ois = new ObjectInputStream(new FileInputStream(filePath))) {
            Employee deserializedEmployee = (Employee) ois.readObject();
            System.out.println("Deserialized Employee object: " + deserializedEmployee);
        } catch (IOException | ClassNotFoundException e) {
            System.out.println("An error occurred during deserialization.");
            e.printStackTrace();
        }
    }
}
```



```
shikhasingh@Shikhas-MacBook-Air Assignment 3.2 %  /usr/bin/env /Users/shikhasingh/Library/Application\ Support/Code/User/globalStorage/pleia
des.java-extension-pack-jdk/java/latest/bin/java --enable-preview -XX:+ShowCodeDetailsInExceptionMessages -cp /Users/shikhasingh/Library/App
lication\ Support/Code/User/workspaceStorage/042383d6666d81e8ac9c92ed8979f8f9/redhat.java/jdt_ws/Assignment\ 3.2_e2a78733/bin SerializationD
emo1
Employee object serialized to employee.ser
Deserialized Employee object: Employee [empId=E123, dept=HR, designation=Manager, Person [id=1, name=Alice, addr=Address [apptDetails=1234 E
lm St, district=Downtown, state=Springfield, country=USA]]]
shikhasingh@Shikhas-MacBook-Air Assignment 3.2 %
```